16444

RECEIVED
JUN 13 1973

Postel

Wayne Hathaway (AMES-67)
30 May 1973

NIC 16444
NWG/RFC 513

COMMENTS ON THE NEW TELNET SPECIFICATIONS

I would like to make the following comments on the proposed new TELNET Protocol Specification (NIC#15372) and TELNET Option Specification (NIC#15373). In general I feel the new TELNET protocol is far superior to the previous version. There are, however, a few items of substance which I feel should be changed, as well as some recommended editorial changes.

I feel the most significant error concerns the "Note on 'Sub-negotiations'" section of the Option Specifications (page 2). The problem stems from the statements "Each party is presumed to be able to parse the parameter(s)" and "Finally, if parameters in an option 'subnegotiation' include a byte with a value of 255, it is not necessary to double this byte in accordance with the general TELNET rules, since the byte cannot be misinterpreted as the TELNET IAC." These two statements make the completely unwarranted but all too prevalent assumption that there is only one "TELNET interpreter" and that all TELNET functions are carried out by it. In particular, problems arise when a TELNET "synch" is received, and the receiving NCP is required to scan the input looking for "interesting" things. If the subnegotiation was in fact being carried out by a user process (and not the "TELNET interpreter") then that user process is probably the only one that knows how to interpret the SB parameters; the NCP itself would have no way of parsing them. As a solution to this problem I propose that all subnegotiation parameters be delimited at the end (perhaps with the same TELNET command SB) and that they be required to obey all TELNET rules, including doubling of IAC's. It may be argued that the user process interpreting the SB itself should do the scanning for "interesting" things, but I do not feel that this burden should be placed on all user processes.

The solution to the above problem is in fact fairly simple to define and implement. The general problem, however, is one of not taking proper cognizance of what I called "user processes" (processes which are not network standards, but which are simply programs attempting to do work using the network). I feel we must be more careful to shape all future network standards with these very real user processes in mind if in fact the network will ever be as useful as is possible.

NIC 16444
NWG/RFC 513

Wayne Hathaway (AMES-67)
30 May 1973

The only editorial comments are on the TELNET Protocol document, which I reference below by page number.

On page 8 the parenthetical comment "(i.e., when a process at one end of a TELNET connection is 'blocked on input')" should either be removed or rewritten to avoid the (to me) incomprehensible phrase "blocked on input." If additional discussion is felt to be necessary, I would propose "(i.e., when a process at one end of a TELNET connection cannot proceed without additional input)." If examples are felt necessary, I would propose "(e.g., in the state characterized by the Multics term 'blocked on input')." The parallel could also be drawn between the GA and the concept of a "read command" being issued to request more input.

On page 10 I feel that there needs to be some more discussion of the Abort Output (AO) command, particularly the statement that it "allows a process ... to run to completion ... but without sending the output to the user's terminal." The problem is that nothing is said about when output is to resume (presumably at the next system prompt character). I realize that the AO is meant only to invoke this function on systems which already provide it, but it would seem to be more useful if more fully specified.

On page 11 I do not understand what the example "(e.g., an overstrike)" is trying to say. Speaking of an overstrike on output would imply to me that both characters are to be printed in the same print position, reducing the EC to a backspace. Some more discussion should also be added as to what EC (and EL) mean on output (if anything).

On page 11 I would like to see the word "promptly" (which is in quotation marks) either eliminated or defined, as per my earlier aside comment. The phrase "if necessary" in the last line also seems unnecessary.

On page 12 my proposed redefinition of "interesting" signals would remove the middle paragraph entirely and would modify the third paragraph substantially. The line "discard all characters which would have had an effect on the NVT printer" should be changed, however, as it seems BELL's should also be discarded.

On page 14 I see no reason why the sequence "CR NUL" is required to generate a "CR" only, and also object to "and the CR character must be avoided in other contexts." Either some supporting discussion should be added or this restriction should be removed.

Wayne Hathaway (AMES-67)
30 May 1973

The second item I object to is the incredibly loose definition of "interesting" things (an aside: words which are so imprecise as to require quotation marks should never appear in protocol specifications). The specifications do define some of these "interesting" things (eg, most TELNET commands) but they then include "and perhaps other characters or character strings as well." To eliminate the difficulty of implementing an undefined set of "interesting" things, I would propose that the set of "interesting" things contain only the DM command itself. The TELNET "synch" would thus be defined as "discard all input up to and including the next DM command." This change should cause no problems with user-generated "interesting" things if they are sent after the "synch" (as specified in the proposed new File Transfer Protocol Specifications). System-generated signals (such as option requests) could be discarded, however, so some additional discussion is in order. If the recommendation that requests not be sent except when something changes is followed, the spontaneous generation of "interesting" things by TELNET itself (whatever that implies) would seem to be rare, especially at the same time that users are generating "synch's". A more positive solution could be had by defining a "synch response" (SR) TELNET command. The SR command would be sent when the INS and DM had both been processed (ie, when the "synch" had been completely received). If a process should ever receive an SR when it has an option request outstanding, the request has been discarded and must be repeated. User processes which carry on option negotiations would be the generators of any "synch's" so they can handle discarded option requests as desired. Note that this assumes that the TELNET process itself will never generate a spontaneous "synch"; comments are solicited on this. Another possible solution would be to define a "TIMING-MARK" TELNET command (see "TELNET Timing Mark Option", NIC#16238), which would be sent immediately following the DM of a "synch". The response to the TIMING-MARK (also to be defined) would mean the same as the proposed SR command.

The final non-editorial comment concerns the need for some means of specifying horizontal tab positions and use. This is quite a nuisance when dealing with systems which normally handle tabs for local terminals. Perhaps this issue can be best handled with an appropriate option; comments are solicited.